(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2003－195749
（P2003－195749A）
(43)公開日　平成15年７月９日(2003.7.9)

(51)Int.Cl.'	識別記号	ＦＩ	ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０９Ｃ　1/00	６１０	Ｇ０９Ｃ　1/00	６１０Ａ　５Ｊ１０４

審査請求　未請求　請求項の数９　ＯＬ　（全 18 頁）

(21)出願番号　特願2001－394109(P2001－394109)

(22)出願日　平成13年12月26日(2001.12.26)

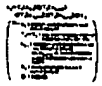
特許法第30条第１項適用申請有り　平成13年９月26日～28日　社団法人情報処理学会開催の「第63回（平成13年後期）全国大会」において文書をもって発表

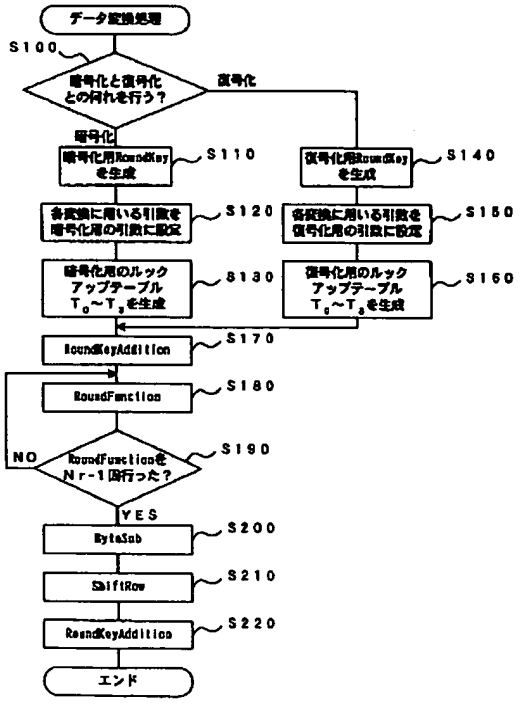
(71)出願人　396021944
株式会社デンソークリエイト
愛知県名古屋市中区錦二丁目15番20号
(71)出願人　301080002
神保　雅一
岐阜県各務原市緑苑南４－112
(72)発明者　陳　志松
愛知県名古屋市中区錦二丁目15番20号　株式会社デンソークリエイト内
(74)代理人　100082500
弁理士　足立　勉

最終頁に続く

(54)【発明の名称】　データ変換装置、データ変換プログラム、記録媒体及びデータ変換方法

(57)【要約】
【課題】　ＡＥＳ暗号方式を用いてデータを暗号化又は復号化する装置に必要なデータ記憶容量を小さくする。
【解決手段】　このデータ変換処理では、ＡＥＳ暗号方式でデータを暗号化又は復号化するために行うＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの４種類の変換について、各変換で用いる引数を、暗号化と復号化とで異なる所定の引数に設定することで（Ｓ１００～Ｓ１６０）、データの暗号化及び復号化を、ＡＥＳ暗号方式の暗号化手順と同じ順序で上記４種類の変換を施すことにより行うようになっている（１７０～Ｓ２２０）。また、Ｓ１８０では、４行４列の行列で表される１２８ビットの処理対象データに上記４種類の変換を順に施すことで得られる処理後データを、下記の式により求めるようになっている。
【数１】





1

【特許請求の範囲】

【請求項１】 ４行４列の行列で表される１２８ビットの変換対象データに、ＡＥＳ暗号方式におけるＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求めるように構成されていることを特徴とするデータ変換装置。

【数１】

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目(i=0,1,2,3)のj列目のデータ(8ビット)

(尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す)

⊕：排他論理和

【請求項２】 ４行４列の行列で表される１２８ビットの変換対象データに、ＡＥＳ暗号方式におけるＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求める機能をコンピュータに実現させるためのデータ変換プログラム。

【数２】

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目(i=0,1,2,3)のj列目のデータ(8ビット)

(尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す)

⊕：排他論理和

【請求項３】 請求項２に記載のデータ変換プログラムが記録されたコンピュータ読み取り可能な記録媒体。

【請求項４】 ４行４列の行列で表される１２８ビットの変換対象データに、ＡＥＳ暗号方式におけるＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求めることを特徴とするデータ変換方法。

【数３】

3 4

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目(i=0,1,2,3)のj列目のデータ(8ビット)

(尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す)

⊕：排他論理和

【請求項５】 ＡＥＳ暗号方式を用いて暗号化されたデータを、下記（ａ１）～（ａ４）の条件に従い復号化することを特徴とするデータ変換方法。

（ａ１）：ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を、ＡＥＳ暗号方式の暗号化手順と同じ順序で行う。

（ａ２）：上記（ａ１）に従い行うＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎでは、暗号化で行われる変換の逆変換となるような引数を用いる。

（ａ３）：上記（ａ１）に従い行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、暗号化で用いられた複数のＲｏｕｎｄＫｅｙを、暗号化と逆の順序で用いる。

（ａ４）：更に、上記（ａ１）に従い行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎのうち、ＭｉｘＣｏｌｕｍｎの次に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、上記（ａ３）に従い用いるＲｏｕｎｄＫｅｙに、復号化で行うＭｉｘＣｏｌｕｍｎの変換を施したものを、ＲｏｕｎｄＫｅｙとして用いる。

【請求項６】 請求項５に記載のデータ変換方法において、

４行４列の行列で表される１２８ビットの変換対象データに、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求めることを特徴とするデータ変換方法。

【数４】

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目(i=0,1,2,3)のj列目のデータ(8ビット)

(尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す)

⊕：排他論理和

【請求項７】 ＡＥＳ暗号方式を用いたデータの暗号化と、ＡＥＳ暗号方式を用いて暗号化されたデータの復号化とを、コンピュータに行わせるためのデータ変換プログラムであって、

データの暗号化及び復号化で共通に用いられると共に、該データに対してＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換をＡＥＳ暗号方式の暗号化手順と同じ順序で行うための共通プログラムと、

前記共通プログラムでの各変換に用いられる引数を、暗号化の場合にはＡＥＳ暗号方式の暗号化手順において用いられる引数に設定し、復号化の場合には下記（ｂ１）

5

～（ｂ３）の条件に従い設定する設定プログラムと、を備えたことを特徴とするデータ変換プログラム。

（ｂ１）：ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎでは、暗号化で行われる変換の逆変換となるような引数が用いられるようにする。

（ｂ２）：ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いられる引数であるＲｏｕｎｄＫｅｙとしては、暗号化で用いられた複数のＲｏｕｎｄＫｅｙが、暗号化と逆の順序で用いられるようにする。

（ｂ３）：更に、ＭｉｘＣｏｌｕｍｎの次に行われるＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いられる引数であるＲｏｕｎｄＫｅｙとしては、上記（ｂ２）に従い用いられるＲｏｕｎｄＫｅｙに、復号化で行われるＭｉｘＣｏｌｕｍｎの変換を施したものが用いられるようにする。

6

【請求項８】　請求項７に記載のデータ変換プログラムにおいて、

前記共通プログラムでは、４行４列の行列で表される１２８ビットの変換対象データに、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求めるようになっていることを特徴とするデータ変換プログラム。

【数５】

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目（j=0,1,2,3）のデータ（32ビット）

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目（i=0,1,2,3）のｊ列目のデータ（8ビット）

（尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す）

$\oplus$：排他論理和

【請求項９】　請求項７又は請求項８に記載のデータ変換プログラムが記録されたコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ＡＥＳ暗号方式を用いてデータを暗号化又は復号化する技術に関する。

【０００２】

【従来の技術】従来より、データを暗号化するための様々な暗号方式が知られている。そして、最近では、米国連邦標準技術局（ＮＩＳＴ）により、ＡＥＳ（Ａｄｖａｎｃｅｄ　Ｅｎｃｒｙｐｔｉｏｎ　Ｓｔａｎｄａｒｄ）と呼ばれる新しい共通暗号方式が発表され、今後の普及が見込まれている。

【０００３】ここで、周知ではあるが、ＡＥＳ暗号方式（以下、単に「ＡＥＳ」ともいう）の概要について説明する。ＡＥＳは、図７に示すように、ブロック長が１２８ビットのブロック暗号である。そして、この１２８ビットのブロックデータは、１バイト（８ビット）単位のデータ（以下、「部分データ」ともいう）に分割され、４×４（４行４列）の正方行列として表現される。尚、図７では、ブロックデータにおけるｉ行目（ｉ＝０，１，２，３）のｊ列目（ｊ＝０，１，２，３）の部分データを、ｂ（ｉ，ｊ）と表している。

【０００４】また、図８に示すように、ＡＥＳに用いられる鍵の鍵長は１２８ビット、１９２ビット、２５６ビットの３種類である。そして、この鍵も、１バイト単位のデータに分割され、４×Ｎｋ（Ｎｋ＝４，６，８）の長方形行列として表現される。

【０００５】そして、ＡＥＳでは、次のような手順でブロックデータを暗号化する。即ち、図９に示すように、まず、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎ（ラウンドキー加算）と呼ばれる変換を１回行う。次に、ＢｙｔｅＳｕｂ（バイト置換）、ＳｈｉｆｔＲｏｗ（行シフト）、ＭｉｘＣｏｌｕｍｎ（列ミックス）及び上記ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの４種類の変換を順に行うＲｏｕｎｄＦｕｎｃｔｉｏｎ（ラウンド変換）と呼ばれる一連の変換を、複数のラウンド（（ラウンド数Ｎｒ）－１回）に渡って繰り返す。そして最後に、ＲｏｕｎｄＦｕｎｃｔｉｏｎの中からＭｉｘＣｏｌｕｍｎだけを除いたＦｉｎａｌＲｏｕｎｄＦｕｎｃｔｉｏｎ（最終ラウンド変換）と呼ばれる変換を、１回行う。尚、ラウンド数Ｎｒは、下記第１表に示すように、鍵長に対応して決められている。

【０００６】

【表１】

第１表

Ｎｋ	4	6	8
Ｎｒ	10	12	14

【０００７】次に、ＡＥＳ暗号方式を用いてデータを暗号化するための上記４種類の変換について、それぞれ説明する。［ＢｙｔｅＳｕｂ］ＢｙｔｅＳｕｂでは、ブロックデータの各単位である各部分データに対して、以下の　，　の計算を行う。

【０００８】　ＧＦ（2^8）における乗法的逆元を計算する（乗算は多項式の積を「$x^8+x^4+x^3+x+1$」でｍｏｄをとる）。

上記　の結果に、各ビットに式（２）のアファイン変換を行う。

【０００９】

【数６】

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \cdots式(2)$$

【００１０】このように、ＢｙｔｅＳｕｂで行う本来の処理は、上記　，　のようなものであるが、本処理では、入力された１バイト長のデータに対して出力を一意に決定できることから、実際には、図１０に示すように、１バイトのデータを１バイトのデータに変換するＳ－ｂｏｘと呼ばれるルックアップテーブルを先に計算しておき、このルックアップテーブルを用いて、各部分データを変換する。

【００１１】つまり、ＢｙｔｅＳｕｂでは、引数であるＳ－ｂｏｘを用いて、ブロックデータを部分データ単位（８ビット単位）で変換するようになっている。

［ＳｈｉｆｔＲｏｗ］ＳｈｉｆｔＲｏｗでは、図１１に示すように、ブロックデータにおける１行目から３行目の各行にそれぞれ対応する引数であるＣ１，Ｃ２，Ｃ３に基づき、各行のデータをバイト単位で循環左シフトする。これにより、各行の部分データは、下記第２表に示すオフセット値だけ左方向へ循環シフトする。

【００１２】

【表２】

第２表

	0行目	1行目	2行目	3行目
オフセット値	0	1	2	3

【００１３】つまり、ＳｈｉｆｔＲｏｗでは、各行に対応する引数Ｃｉを用いて、ブロックデータを行単位（３２ビット単位）で変換するようになっている。

［ＭｉｘＣｏｌｕｍｎ］ＭｉｘＣｏｌｕｍｎでは、ブロックデータにおける各列に対して、ＧＦ（2^8）において、各部分データを係数とする多項式と下記の式（３）との乗算を行い、その積を「x^4+1」でｍｏｄをとる。

【００１４】

$$c(x)='03'x^3+'01'x^2+'01'x+'02' \quad \cdots式(3)$$

そして、得られた多項式の係数を対応する各部分データの値とする。つまり、ＭｉｘＣｏｌｕｍｎでは、図１２に示すように、ブロックデータにおける各列のデータに対して行列Ｃをかけ算することで変換する（即ち、引数である行列Ｃを用いて、ブロックデータを列単位（３２ビット単位）で変換する）ようになっている。

【００１５】［ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎ］ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、引数としてＲｏｕｎｄＫｅｙ（ラウンドキー）と呼ばれる鍵を用いる。そして、このＲｏｕｎｄＫｅｙは、前述した元々の鍵（図８）から生成される。

【００１６】即ち、図１３に示すように、元々の鍵を拡張アルゴリズムに従って拡張し、その拡張した鍵をブロックデータのブロック長（１２８ビット）で順番に区切ることにより、図９に示したブロックデータの暗号化手順に含まれるＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの回数分（（ラウンド数Ｎｒ）＋１回分）のＲｏｕｎｄＫｅｙを生成する。こうして生成した複数のＲｏｕｎｄＫｅｙ

9

は、データの暗号化のために複数回行われるＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで順に用いられる。

【００１７】そして、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、図１４に示すように、ブロックデータとＲｏｕｎｄＫｅｙとのＥＸＯＲ（排他論理和）をビット単位でとる。つまり、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、引数であるＲｏｕｎｄＫｅｙを用いて、ブロックデータを１ビット単位で変換するようになっている。

【００１８】以上説明した４種類の変換からなるＲｏｕｎｄＦｕｎｃｔｉｏｎの処理のイメージを図１５に示す。ところで、こうしたデータの暗号化処理では、処理速度を向上させるための手法として、ルックアップテーブルが一般に用いられる。

【００１９】ここで、ＡＥＳでは、１２８ビットのブロックデータ毎に変換が行われるため、これを１つのルックアップテーブルを用いて変換しようとすると、「2^{128}×１６バイト」といった膨大な大きさのルックアップテーブルが必要となってしまい、現実的ではない。

【００２０】そこで、本発明者は、ＡＥＳ暗号方式の特徴に着目して、次のようなルックアップテーブルを考えた。即ち、ＲｏｕｎｄＦｕｎｃｔｉｏｎでは、行単位の変換であるＳｈｉｆｔＲｏｗと、列単位の変換であるＭｉｘＣｏｌｕｍｎとが行われることで、結果として１２８ビット単位で変換が行われるようになっているが、ＳｈｉｆｔＲｏｗでは、部分データの位置を移動させているだけであるため、この点を考慮すれば、ＲｏｕｎｄＦｕｎｃｔｉｏｎ１回分の処理については、列単位（３２ビット単位）の変換であるとみなすことができる。つまり、ルックアップテーブルを用いて変換する３２ビットのデータとして、ＳｈｉｆｔＲｏｗの変換を施すことにより同じ列に移動する４つの部分データを予め選択することで、ＳｈｉｆｔＲｏｗを考慮する必要がなくなるのである。

【００２１】このため、列単位（３２ビット単位）で変換を行うルックアップテーブルを用いてデータを変換することが可能となる。

10

【００２２】

【発明が解決しようとする課題】しかしながら、列単位のルックアップテーブルであっても、「2^{32}×４バイト＝１６３８４メガバイト」といった非常に大きなものとなるため、データを暗号化する装置にも非常に大きなデータ記憶容量が必要になってしまう。

【００２３】一方また、こうした問題に加え、データの暗号化と復号化との両方を行うためのプログラムでは、暗号化のみを行うためのプログラムに比べ、更に大きなデータ記憶容量が必要となってしまう。ＡＥＳ暗号方式を用いて暗号化されたデータは、暗号化の処理と全く逆の処理を行うことで復号化することができるが、この場合には、暗号化と復号化とで２種類のプログラムが必要となってしまうからである。

【００２４】以上のように、暗号化や復号化といったデータ変換処理を行う装置に大きなデータ記憶容量が必要になると、特に携帯電話装置のような小型機器への実装が困難になる。本発明は、こうした問題に鑑みなされたものであり、ＡＥＳ暗号方式を用いてデータを暗号化又は復号化する装置に必要なデータ記憶容量を小さくすることを目的としている。

【００２５】

【課題を解決するための手段及び発明の効果】上記目的を達成するためになされた請求項１に記載のデータ変換装置は、４行４列の行列で表される１２８ビットの変換対象データに、ＡＥＳ暗号方式におけるＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、下記式（１）により求めるように構成されている。

【００２６】

【数７】

$$h_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \quad \cdots式(1)$$

但し、h_j：変換後データにおけるj列目（j=0,1,2,3）のデータ（32ビット）

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：変換対象データにおけるi行目（i=0,1,2,3）のj列目のデータ（8ビット）

（尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す）

⊕：排他論理和

【００２７】つまり、請求項１のデータ変換装置では、変換対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及

びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを上記式（１）により求める、といった請求項４のデータ変換方法を用いている。

【００２８】ここで、各ルックアップテーブルT_0～T_3の大きさは、それぞれ「2^8×４バイト＝１０２４バイト」であるから、４つのルックアップテーブルを合わせても４０９６バイトである。この値は、前述した列単位で変換を行うルックアップテーブルの大きさ（１６３８４メガバイト）に比べ、格段に小さい。

【００２９】このような請求項１のデータ変換装置によれば、ＡＥＳ暗号方式を用いたデータの暗号化の処理速度を向上させつつ、ＡＥＳ暗号方式を用いてデータを暗号化する装置に必要なデータ記憶容量を小さくすることができる。即ち、ＡＥＳ暗号方式を用いたデータの暗号化では、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に行う処理を複数回繰り返すようになっているが、この処理で得られる値を本データ変換装置（請求項４のデータ変換方法）により求めることで、各変換を順に行う必要が無くなり、処理速度を向上させることができる。しかも、使用するルックアップテーブルが小さいことから、ＡＥＳ暗号方式を用いてデータを暗号化する装置に必要なデータ記憶容量を小さくすることができる。

【００３０】次に、請求項２に記載のデータ変換プログラムは、４行４列の行列で表される１２８ビットの変換対象データに、ＡＥＳ暗号方式におけるＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、上記式（１）により求める機能をコンピュータに実現させるためのものである。

【００３１】つまり、請求項２のデータ変換プログラムでも、変換対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、請求項４のデータ変換方法を用いて求めるようになっている。

【００３２】このような請求項２のデータ変換プログラムを、ＡＥＳ暗号方式の暗号化を行うために用いれば、データの暗号化の処理速度を向上させつつ、ＡＥＳ暗号方式を用いてデータを暗号化する装置に必要なデータ記憶容量を小さくすることができる。また、請求項２のデータ変換プログラムは、請求項３のようにコンピュータ読み取り可能な記録媒体に記録されていてもよい。

【００３３】次に、請求項５に記載のデータ変換方法は、ＡＥＳ暗号方式を用いて暗号化されたデータを、下記（ａ１）～（ａ４）の条件に従い復号化することを特徴としている。

（ａ１）：ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を、ＡＥＳ暗号方式の暗号化手順と同じ順序で行う。

【００３４】（ａ２）：上記（ａ１）に従い行うＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎでは、暗号化で行われる変換の逆変換となるような引数を用いる。

（ａ３）：上記（ａ１）に従い行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、暗号化で用いられた複数のＲｏｕｎｄＫｅｙを、暗号化と逆の順序で用いる。

【００３５】（ａ４）：更に、上記（ａ１）に従い行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎのうち、ＭｉｘＣｏｌｕｍｎの次に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、上記（ａ３）に従い用いるＲｏｕｎｄＫｅｙに、復号化で行うＭｉｘＣｏｌｕｍｎの変換を施したものを、ＲｏｕｎｄＫｅｙとして用いる。

【００３６】つまり、請求項５のデータ変換方法では、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換に、暗号化の場合とは異なる引数を用いることで、ＡＥＳ暗号方式の暗号化手順と同じ順序で各変換を行いつつデータを復号化するようになっている。

【００３７】このような請求項５のデータ変換方法を、ＡＥＳ暗号方式の暗号化及び復号化を行うためのプログラムに用いれば、そのプログラムサイズを小さくすることができる。つまり、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を暗号化手順と同じ順序で行うためのプログラムを共用できるからである。

【００３８】次に、上記請求項５のデータ変換方法を用いた請求項７に記載のデータ変換プログラムは、ＡＥＳ暗号方式を用いたデータの暗号化と、ＡＥＳ暗号方式を用いて暗号化されたデータの復号化とを、コンピュータに行わせるためのものである。

【００３９】そして、このデータ変換プログラムは、データの暗号化及び復号化で共通に用いられると共に、そのデータに対してＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換をＡＥＳ暗号方式の暗号化手順と同じ順序で行うための共通プログラムと、この共通プログラムでの各変換に用いられる引数を、暗号化の場合にはＡＥＳ暗号方式の暗号化手順において用いられる引数に設定し、復号化の場合には下記（ｂ１）～（ｂ３）の条件に従い設定する設定プログラムと、を備えている。

【００４０】（ｂ１）：ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎでは、暗号化で行われる変換の逆変換となるような引数が用いられるようにする。

（ｂ２）：ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いられる引数であるＲｏｕｎｄＫｅｙとしては、暗号化で用いられた複数のＲｏｕｎｄＫｅｙが、暗号化と逆の順序で用いられるようにする。

【００４１】（ｂ３）：更に、ＭｉｘＣｏｌｕｍｎの次に行われるＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いら

れる引数であるＲｏｕｎｄＫｅｙとしては、上記（ｂ２）に従い用いられるＲｏｕｎｄＫｅｙに、復号化で行われるＭｉｘＣｏｌｕｍｎの変換を施したものが用いられるようにする。

【００４２】このような請求項７のデータ変換プログラムは、データを暗号化するためのプログラムとデータを復号化するためのプログラムとを独立して備えるものに比べ、プログラムサイズが格段に小さい。したがって、このデータ変換プログラムによれば、ＡＥＳ暗号方式を用いてデータの暗号化及び復号化を行う装置に必要なデータ記憶容量を小さくすることができる。

【００４３】ところで、請求項４のデータ変換方法によれば、変換対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを上記式（１）により求めることで、データの暗号化の処理速度を向上させつつ、ＡＥＳ暗号方式を用いてデータを暗号化する装置に必要なデータ記憶容量を小さくすることができる。一方、上記請求項５のデータ変換方法ではデータを復号化する場合に、また上記請求項７のデータ変換プログラムではデータを復号化する場合にも、ＡＥＳ暗号方式の暗号化手順と同じ順序でＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎの各変換を行うこととなる。よって、上記請求項４のデータ変換方法は、請求項６又は請求項８に記載のように、請求項５と請求項７とのそれぞれにも適用することができる。

【００４４】即ち、請求項６に記載のデータ変換方法では、上記請求項５のデータ変換方法において、４行４列の行列で表される１２８ビットの変換対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、上記式（１）により求めることを特徴としている。

【００４５】また、請求項８に記載のデータ変換プログラムでは、上記請求項７のデータ変換プログラムにおいて、共通プログラムが、４行４列の行列で表される１２８ビットの変換対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られる変換後データを、上記式（１）により求めるようになっている。

【００４６】そして、上記請求項６のデータ変換方法によれば、ＡＥＳ暗号方式を用いて暗号化されたデータの復号化の処理速度を向上させつつ、ＡＥＳ暗号方式を用いてデータを復号化する装置に必要なデータ記憶容量を小さくすることができる。更に、データの暗号化と復号化との両方をコンピュータに行わせるためのデータ変換プログラムに適用すれば、請求項５の効果に加え、データの暗号化と復号化との両方の処理速度を向上させることができる。

【００４７】また、上記請求項８のデータ変換プログラムによれば、請求項７の効果に加え、データの暗号化及び復号化の処理速度を向上させつつ、ＡＥＳ暗号方式を用いてデータの暗号化及び復号化を行う装置に必要なデータ記憶容量を小さくすることができる。

【００４８】尚、請求項７，８のデータ変換プログラムは、請求項９のようにコンピュータ読み取り可能な記録媒体に記録されていてもよい。

【００４９】

【発明の実施の形態】以下、本発明が適用された実施形態のデータ変換装置としての携帯電話装置について、図面を用いて説明する。まず図１は、本実施形態の携帯電話装置１０の構成を表すブロック図である。

【００５０】この携帯電話装置１０は、送受信部１２、ＣＰＵ１４、ＲＡＭ１６、ＲＯＭ１８及びデータの書き換えが可能な不揮発性メモリ２０を備えている。送受信部１２は、無線通信によりデータの送受信を行う。また、不揮発性メモリ２０には、外部のデータベースとの間でデータをやりとりするデータ送受信処理をＣＰＵ１４に行わせるためのアプリケーションプログラムと、このデータ送受信処理で送受信されるデータをＡＥＳ暗号方式を用いて暗号化又は復号化する処理（以下、「データ変換処理という」）をＣＰＵ１４に行わせるためのデータ変換プログラムとが記憶されている。

【００５１】ここで、上記データ変換プログラムに従いＣＰＵ１４が行うデータ変換処理について、図２のフローチャートを用いて説明する。尚、本データ変換処理は、上記データ送受信処理にて平文データ（即ち、暗号化されていないデータ）を外部のデータベースへ送信しようとする場合又は外部のデータベースから暗号データを受信した場合に開始される。

【００５２】このデータ変換処理が開始されると、まずＳ１００にて、暗号化と復号化との何れの処理を行うのかを判定する。具体的には、上記データ送受信処理にて平文データを送信しようとする場合には、暗号化を行うと判定し、上記データ送受信処理にて暗号データを受信した場合には、復号化を行うと判定する。

【００５３】そして、Ｓ１００で、暗号化を行うと判定した場合には、Ｓ１１０へ移行し、暗号化用のＲｏｕｎｄＫｅｙを生成して、ＲＡＭ１６に記憶させる。ここで、暗号化用のＲｏｕｎｄＫｅｙとは、平文データを暗号化する際に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙのことであり、従来技術（図１３）で説明したように、鍵を拡張して順番に区切ることにより複数生成する。

【００５４】このＳ１１０で暗号化用のＲｏｕｎｄＫｅｙを生成した後は、Ｓ１２０へ移行し、後述するＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換に用いる引数を、暗号化用の引数に設定する。具体的には、従来技術で説明したように、ＢｙｔｅＳｕｂで用いる引数を、図１０で説明したＳ－ｂｏｘに設定する。また、ＳｈｉｆｔＲｏｗで用いる引

15

数を、図１１で説明したＣｉに設定する。更に、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いる引数を、図１４で説明したＲｏｕｎｄＫｅｙ（即ち、Ｓ１１０にて生成した暗号化用のＲｏｕｎｄＫｅｙ）に設定する。

【００５５】そして次に、Ｓ１３０へ移行し、後述するＲｏｕｎｄＦｕｎｃｔｉｏｎで用いる暗号化用の４つのルックアップテーブルT_0～T_3を生成して、ＲＡＭ１６に記憶させる。一方、Ｓ１００で、復号化を行うと判定した場合には、Ｓ１４０へ移行し、復号化用のＲｏｕｎｄＫｅｙを生成して、ＲＡＭ１６に記憶させる。ここで、復号化用のＲｏｕｎｄＫｅｙとは、暗号データを復号化する際に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙのことであり、次の（ｃ１），（ｃ２）の条件を満たすものである。

【００５６】（ｃ１）：復号化で行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎでは、暗号化で用いられた複数のＲｏｕｎｄＫｅｙを、暗号化と逆の順序で用いる。

（ｃ２）：但し、（ｃ１）の条件に従い用いるＲｏｕｎｄＫｅｙのうち、ＲｏｕｎｄＦｕｎｃｔｉｏｎで行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙについては、更に、暗号化で行うＭｉｘＣｏｌｕｍｎの逆変換を施したものをＲｏｕｎｄＫｅｙとして用いる。

【００５７】したがって、復号化用のＲｏｕｎｄＫｅｙは、次のように生成される。まず、暗号化の場合と同様に、鍵を拡張して順番に区切ることにより複数のＲｏｕｎｄＫｅｙ（即ち、暗号化で用いられるＲｏｕｎｄＫｅｙ）を生成する。次に、生成した複数のＲｏｕｎｄＫｅｙを、暗号化とは逆の順序で各ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎに割り振る。更に、ＲｏｕｎｄＦｕｎｃｔｉｏｎで用いるＲｏｕｎｄＫｅｙについては、暗号化で行うＭｉｘＣｏｌｕｍｎの逆変換を施す。

【００５８】こうしてＳ１４０で復号化用のＲｏｕｎｄＫｅｙを生成した後は、Ｓ１５０へ移行し、後述するＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換に用いる引数を、復号化用の引数に設定する。具体的には、ＢｙｔｅＳｕｂで用いる

16

引数を、暗号化で用いるＳ－ｂｏｘの逆変換となるルックアップテーブルに設定する。また、ＳｈｉｆｔＲｏｗで用いる引数を、暗号化で用いるＣｉの逆変換となる引数（即ち、ブロックデータにおける各行の部分データを暗号化の場合と同じオフセット値だけ右方向へ循環シフトする引数）に設定する。更に、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いる引数を、Ｓ１４０にて生成した復号化用のＲｏｕｎｄＫｅｙに設定する。

【００５９】そして次に、Ｓ１６０へ移行し、後述するＲｏｕｎｄＦｕｎｃｔｉｏｎで用いる復号化用の４つのルックアップテーブルT_0～T_3を生成して、ＲＡＭ１６に記憶させる。こうして、Ｓ１３０又はＳ１６０でルックアップテーブルT_0～T_3を生成した後は、以下のＳ１７０～Ｓ２２０にて、ＡＥＳ暗号方式の暗号化手順と同じ順序でＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を行うようになっている。尚、Ｓ１００～Ｓ１６０の処理をＣＰＵ１４に行わせるためのプログラムが、設定プログラムに相当し、Ｓ１７０～Ｓ２２０の処理をＣＰＵ１４に行わせるためのプログラムが、共通プログラムに相当する。

【００６０】まず、Ｓ１７０では、暗号データ又は平文データであるブロックデータに対して、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの変換を１回行う。次に、Ｓ１８０へ移行し、ＲｏｕｎｄＦｕｎｃｔｉｏｎを行う。ここで、本データ変換処理におけるＲｏｕｎｄＦｕｎｃｔｉｏｎの処理について説明する。

【００６１】ＲｏｕｎｄＦｕｎｃｔｉｏｎは、ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を順に施す処理であるが、このＳ１８０では、各変換をそれぞれ行うのではなく、４行４列の行列で表される１２８ビットの処理対象データ（変換対象データに相当）にＲｏｕｎｄＦｕｎｃｔｉｏｎを行うことで得られる処理後データを、下記式（４）により求めるようになっている。

【００６２】

【数８】

17 18

$$e_j = T_0[a_{0,j}] \oplus T_1[a_{1,j-C1}] \oplus T_2[a_{2,j-C2}] \oplus T_3[a_{3,j-C3}] \oplus k_j \quad \cdots式(4)$$

但し、e_j：処理後データにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$T_0 \sim T_3$：8ビットのデータを32ビットのデータに変換するルックアップテーブル

$a_{i,j}$：処理対象データにおけるi行目(i=0,1,2,3)のj列目のデータ(8ビット)

(尚、$a_{i,j-Ci}$ は、ShiftRowの変換を施すことによりi行目のj列目へ移動するデータを表す)

k_j：RoundKeyにおけるj列目(j=0,1,2,3)のデータ(32ビット)

$\oplus$：排他論理和

【００６３】上記式（４）では、処理対象データから処理後データへの変換を３２ビット単位で行う。具体的には、図３に示すように、処理対象データを構成する部分データ（８ビット）の中から、ＳｈｉｆｔＲｏｗの変換を施すことにより同列に移動する４つの部分データ（$a_{0,j}$、$a_{1,j-C1}$、$a_{2,j-C2}$、$a_{3,j-C3}$）を選択し、この選択した４つの部分データからなる３２ビットのデータを、処理後データにおける同列の４つの部分データからなる３２ビットのデータe_jに変換するようになっている。

【００６４】また、上記式（４）では、処理対象データにＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎの各変換を順に施すことで得られるデータ（変換後データに相当）を、４つのルックアップテーブル$T_0 \sim T_3$を用いて求めるようになっている。具体的には、処理対象データから選択した各部分データ（$a_{0,j}$、$a_{1,j-C1}$、$a_{2,j-C2}$、$a_{3,j-C3}$）を、対応するルックアップテーブル$T_0 \sim T_3$を用いてそれぞれ変換し、この変換後の値の排他論理和をとることで求めている。そして更に、このように求めたデータに、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの変換（即ち、上記式（４）におけるk_jの加算）を施して、処理後データを求めるようになっている。尚、このＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの変換が、ＭｉｘＣｏｌｕｍｎの次に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎに相当する。

【００６５】ここで、各ルックアップテーブル$T_0 \sim T_3$は、処理対象データの部分データ（８ビット）を３２ビットのデータに変換するものである。したがって、各ルックアップテーブル$T_0 \sim T_3$の大きさは、それぞれ「$2^8 \times 4$バイト＝１０２４バイト」であり、４つのルックアップテーブルを合わせても４０９６バイトである。

【００６６】そして、このようなＳ１８０の処理を行った後はＳ１９０へ移行し、Ｓ１８０の処理をＮｒ－１回行ったか否かを判定する。尚、ラウンド数Ｎｒは、前述したように、第１表に示す値である。このＳ１９０で、Ｎｒ－１回行っていないと判定した場合には、Ｓ１８０へ戻る。つまり、Ｓ１８０の処理をＮｒ－１回繰り返すようになっている。

【００６７】一方、Ｓ１９０で、Ｎｒ－１回行ったと判定した場合には、Ｓ２００へ移行してＢｙｔｅＳｕｂの変換を行い、次に、Ｓ２１０へ移行してＳｈｉｆｔＲｏｗの変換を行い、更に、Ｓ２２０へ移行してＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの変換を行い、本データ変換処理を終了する。

【００６８】以上説明したように、上記データ変換処理では、平文データを暗号化する場合だけでなく、暗号データを復号化する場合にも、ＡＥＳ暗号方式の暗号化手順と同じ順序でＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を行うようになっている。ここで、暗号データの復号化を平文データの暗号化と同じ変換順序で行うことができる理由について説明する。

【００６９】そもそもＡＥＳ暗号方式を用いて暗号化された暗号データは、暗号化の処理と全く逆の処理を行う（即ち、暗号化で行う各変換の逆変換を暗号化とは逆の順序で行う）ことで復号化することができる。ＲｏｕｎｄＦｕｎｃｔｉｏｎを例にして説明すると、図４に示すように、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎ→ＩｎｖＭｉｘＣｏｌｕｍｎ→ＩｎｖＳｈｉｆｔＲｏｗ→ＩｎｖＢｙｔｅＳｕｂの順に変換を行えばよい。

【００７０】ここで、ＩｎｖＭｉｘＣｏｌｕｍｎ，ＩｎｖＳｈｉｆｔＲｏｗ，ＩｎｖＢｙｔｅＳｕｂは、それぞれ、暗号化の場合のＭｉｘＣｏｌｕｍｎ，ＳｈｉｆｔＲｏｗ，ＢｙｔｅＳｕｂの逆変換であり、変換に用いる引数が異なるだけである。即ち、ＩｎｖＭｉｘＣｏｌｕｍｎでは、暗号化で用いる行列Ｃの逆変換となる行列を引

数として用いてデータを変換する。また、ＩｎｖＳｈｉｆｔＲｏｗでは、暗号化で用いるＣｉの逆変換となる引数を用いてデータを変換する。また更に、ＩｎｖＢｙｔｅＳｕｂでは、暗号化で用いるＳ－ｂｏｘの逆変換となるルックアップテーブルを引数として用いてデータを変換する。尚、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎについては、ＲｏｕｎｄＫｅｙの排他論理和をとる変換であるため、暗号化と同じＲｏｕｎｄＫｅｙを用いて変換すれば、逆変換となる。

【００７１】そして、ＩｎｖＳｈｉｆｔＲｏｗ→ＩｎｖＢｙｔｅＳｕｂという変換順序は、ＩｎｖＢｙｔｅＳｕｂ→ＩｎｖＳｈｉｆｔＲｏｗという順序に入れ替えても、結果は変わらない。また、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎ→ＩｎｖＭｉｘＣｏｌｕｍｎという変換順序は、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙにＩｎｖＭｉｘＣｏｌｕｍｎの変換を施せば、ＩｎｖＭｉｘＣｏｌｕｍｎ→ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎという順序に入れ替えても、結果は変わらない。

【００７２】即ち、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙを「Ｋ」、ＩｎｖＭｉｘＣｏｌｕｍｎの変換行列を「Ｄ」、変換の対象となるデータを「Ｂ」とすると、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎ→ＩｎｖＭｉｘＣｏｌｕｍｎという復号化処理は、下記式（５）の左辺のように表すことができ、更に、右辺のように展開することができる。

【００７３】（Ｂ＋Ｋ）・Ｄ＝Ｂ・Ｄ＋Ｋ・Ｄ　…式（５）

そのため、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで、ＲｏｕｎｄＫｅｙにＩｎｖＭｉｘＣｏｌｕｍｎの変換を施して得られる値「Ｋ・Ｄ」をＲｏｕｎｄＫｅｙとして用いれば、ＩｎｖＭｉｘＣｏｌｕｍｎ→ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎという順序に入れ替えることができる。

【００７４】その結果、図５の左側に示すような復号化の変換順序（即ち、暗号化手順と逆の順序）を、ＩｎｖＳｈｉｆｔＲｏｗとＩｎｖＢｙｔｅＳｕｂとの順序を入れ替え、更に、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎとＩｎｖＭｉｘＣｏｌｕｍｎとの順序を入れ替えることで、図５の右側に示すように、暗号化手順と同じ順序に変更することができる。

【００７５】以上の理由から、各変換に用いる引数を、下記（ｄ１）～（ｄ３）の条件に従い設定すれば、暗号化手順と同じ変換順序で復号化を行うことができる。

（ｄ１）：ＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ及びＭｉｘＣｏｌｕｍｎでは、暗号化で行われる変換の逆変換となるような引数を用いる。

【００７６】（ｄ２）：ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙとしては、暗号化で用いられた複数のＲｏｕｎｄＫｅｙを、暗号化と逆の順序で用いる。

（ｄ３）：更に、ＭｉｘＣｏｌｕｍｎの次に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いるＲｏｕｎｄＫｅｙとしては、上記（ｄ２）に従い用いるＲｏｕｎｄＫｅｙに、復号化で行うＭｉｘＣｏｌｕｍｎの変換を施したものを用いる。

【００７７】そして、前述したデータ変換処理では、こうした順序変更に加え、更に、ＲｏｕｎｄＦｕｎｃｔｉｏｎを上記式（４）により行うようになっている。即ち、暗号化の場合には、処理対象データに対して暗号化用の引数を用いたＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を順に施すことで得られる処理後データを、式（４）により求めるようになっており、言い換えれば、このような処理後データが得られるような暗号化用のルックアップテーブルT_0～T_3をＳ１３０にて生成するようになっている。

【００７８】また、復号化の場合には、処理対象データに対して復号化用の引数（即ち、上記（ｄ１）～（ｄ３）の条件を満たす引数）を用いたＢｙｔｅＳｕｂ、ＳｈｉｆｔＲｏｗ、ＭｉｘＣｏｌｕｍｎ及びＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの各変換を順に施すことで得られる処理後データを、式（４）により求めるようになっており、言い換えれば、このような処理後データが得られるような復号化用のルックアップテーブルT_0～T_3をＳ１６０にて生成するようになっている。

【００７９】次に、本発明者が行った試験内容について簡単に説明する。本発明者は、ＡＥＳ暗号方式を用いてデータの暗号化及び復号化を行う上記実施形態のデータ変換プログラムを市販の携帯電話装置へ実装し、動作の確認を行った。

【００８０】今回、実装の対象とした携帯電話装置は、Ｊａｖａ（登録商標）アプリケーションを実行可能な機種であり、Ｊａｖａ（登録商標）アプリケーションの大きさについて、Ｊａｒファイル（Ｊａｖａ（登録商標）の圧縮ファイル）で１０キロバイト以内という制約事項があった。本発明者は、実際にＡＥＳ暗号クラス（上記実施形態のデータ変換プログラム）を作成して、その作成したＡＥＳ暗号クラスをＪａｒファイルに圧縮した。その結果、暗号クラス（圧縮後のプログラム）の大きさは２．４キロバイトとなり、この携帯電話装置に実装することができた。

【００８１】そして、図６に示すように、この携帯電話装置から外部の就職先データベースへアクセスするアプリケーションを開発し、送信データの暗号化及び受信データの復号化を実際に行えることを確認した。このような本実施形態の携帯電話装置１０によれば、ＡＥＳ暗号方式を用いたデータの暗号化及び復号化を、速い処理速度で、しかも比較的小さなデータ記憶容量で実現することができる。

【００８２】ここで、暗号化及び復号化の処理速度が速

21

くなる理由としては、データ変換処理において、処理対象データにＲｏｕｎｄＦｕｎｃｔｉｏｎの処理を行うことで得られる処理後データを上記式（４）により求めるといったデータ変換方法が用いられていることが挙げられる。

【００８３】一方、データ記憶容量を小さくすることができる理由としては、データ変換処理をＣＰＵ１４に行わせるためのデータ変換プログラムのプログラムサイズが小さいことが挙げられる。即ち、データ変換処理において、暗号化と復号化とを同一の変換順序で行うといったデータ変換方法が用いられており、しかも、上記式（４）で使用するルックアップテーブルT_0～T_3を、データ変換処理の中で生成するようになっているからである。

【００８４】また、データ記憶容量が小さくできるもう一つの理由としては、上記式（４）で使用するルックアップテーブルT_0～T_3が小さいことが挙げられる。以上、本発明の一実施形態について説明したが、本発明は、種々の形態を採り得ることは言うまでもない。

【００８５】例えば、上記実施形態の携帯電話装置１０では、データ変換処理を開始してからルックアップテーブルT_0～T_3を生成するようになっているが、これに限らず、データ変換プログラムの一部としてルックアップテーブルT_0～T_3のデータを予め記憶していてもよい。このようにすれば、ルックアップテーブルT_0～T_3を生成する必要が無くなる分、処理速度をより速くすることができる。

【００８６】また、上記実施形態の携帯電話装置１０では、データを暗号化又は復号化する際に行うＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎで用いる複数のＲｏｕｎｄＫｅｙを一度に生成するようになっているが、これに限らず、ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎを行う毎に、そこで必要となるＲｏｕｎｄＫｅｙのみを生成するようにし、不要になったＲｏｕｎｄＫｅｙについてはすぐに消去するようにしてもよい。このようにすれば、ＲＡＭ１６に必要なデータ容量をより小さくすることができる。

【００８７】一方、上記実施形態では、データの暗号化及び復号化を行うデータ変換装置としての携帯電話装置１０について述べたが、データ変換プログラム自体や、

22

このデータ変換プログラムが記録されたコンピュータ読み取り可能な記録媒体（例えば、フロッピー（登録商標）ディスクやＣＤ－ＲＯＭ等）や、データ変換方法自体も、本発明の範囲である。

【図面の簡単な説明】

【図１】　実施形態の携帯電話装置の構成を表すブロック図である。

【図２】　データ変換処理を表すフローチャートである。

【図３】　データ変換処理におけるＲｏｕｎｄＦｕｎｃｔｉｏｎの処理のイメージを表す説明図である。

【図４】　データを復号化する場合のＲｏｕｎｄＦｕｎｃｔｉｏｎの処理のイメージを表す説明図である。

【図５】　データを復号化する場合の変換順序を表す説明図である。

【図６】　データ変換プログラムを市販の携帯電話装置へ実装した結果を説明する説明図である。

【図７】　ＡＥＳ暗号方式に用いられるブロックデータを説明する説明図である。

【図８】　ＡＥＳ暗号方式に用いられる鍵を説明する説明図である。

【図９】　ＡＥＳ暗号方式の暗号化手順を説明する説明図である。

【図１０】　ＢｙｔｅＳｕｂの変換を説明する説明図である。

【図１１】　ＳｈｉｆｔＲｏｗの変換を説明する説明図である。

【図１２】　ＭｉｘＣｏｌｕｍｎの変換を説明する説明図である。

【図１３】　ＲｏｕｎｄＫｅｙの生成方法を説明する説明図である。

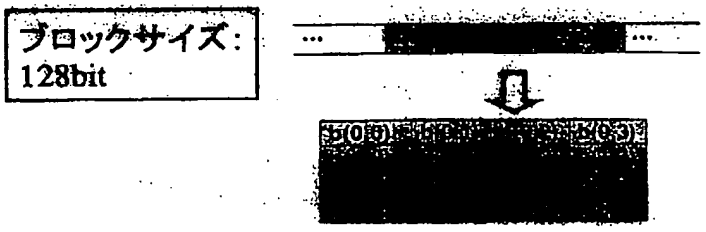
【図１４】　ＲｏｕｎｄＫｅｙＡｄｄｉｔｉｏｎの変換を説明する説明図である。

【図１５】　ＲｏｕｎｄＦｕｎｃｔｉｏｎの処理のイメージを表す説明図である。

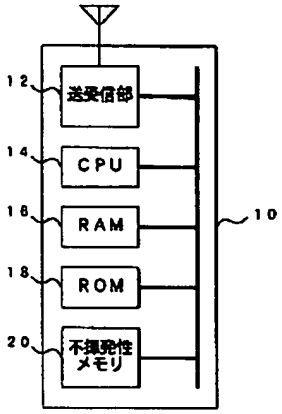
【符号の説明】

１０…携帯電話装置、１２…送受信部、１４…ＣＰＵ、１６…ＲＡＭ、１８…ＲＯＭ、２０…不揮発性メモリ

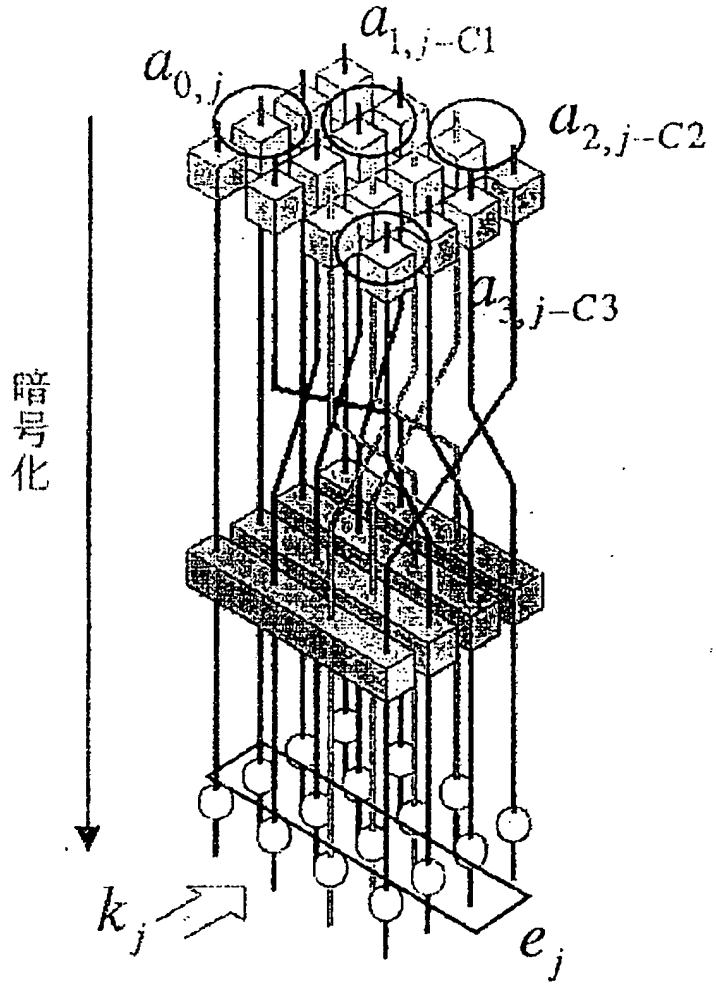
【図７】



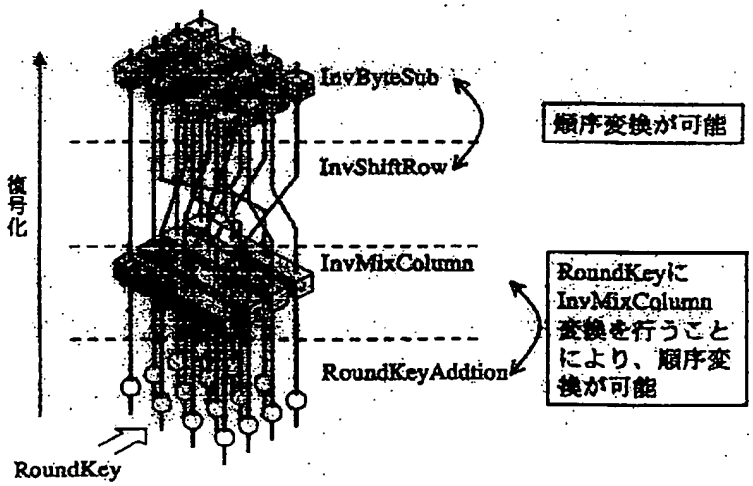
【図１】


12
送受信部
14
CPU
16
RAM
18
ROM
20
不揮発性メモリ
10

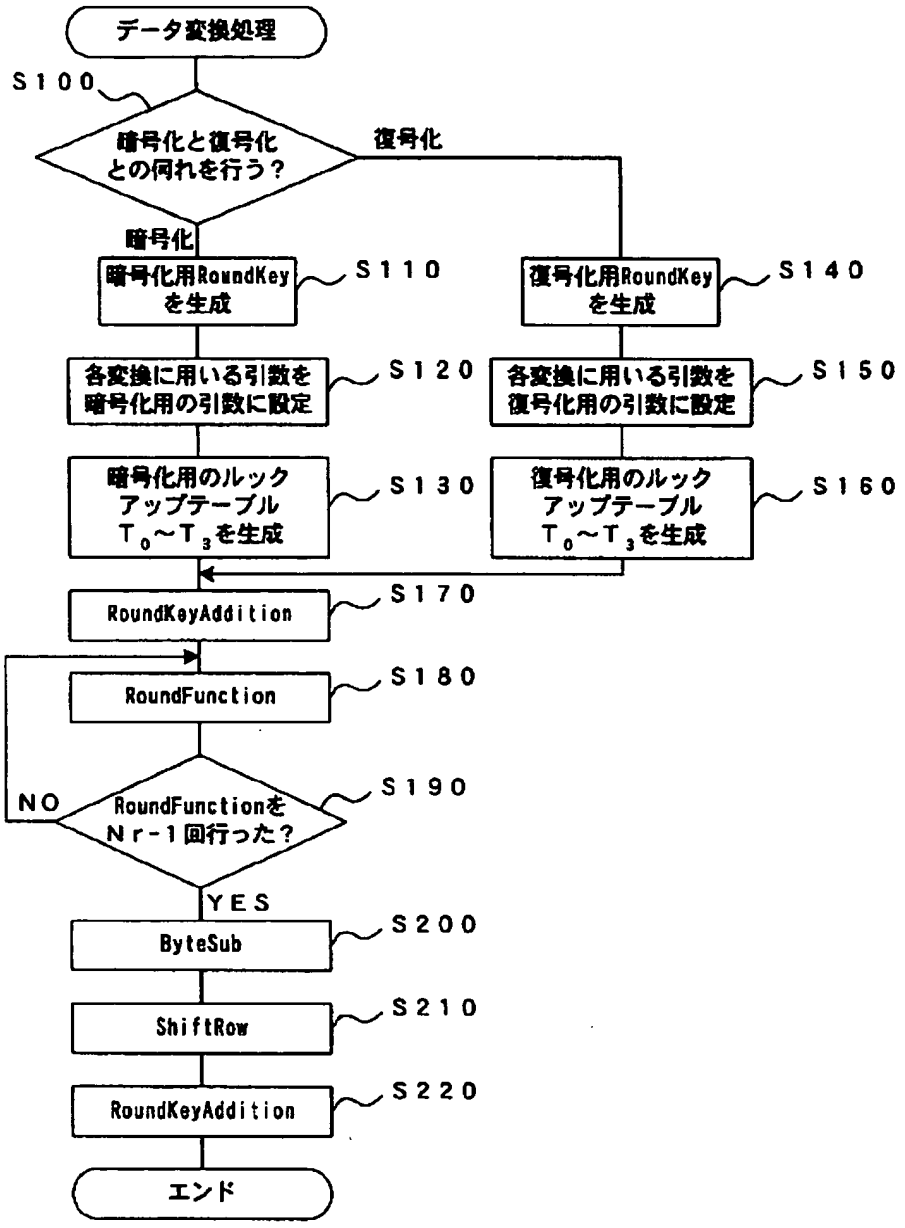
【図３】


$a_{1,j-C1}$
$a_{0,j}$
$a_{2,j-C2}$
$a_{3,j-C3}$
暗号化
k_j
e_j

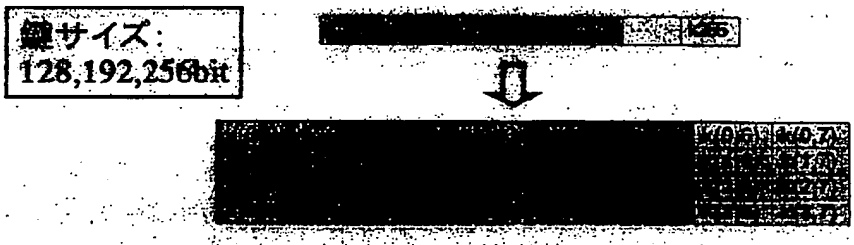
【図４】


InvByteSub
InvShiftRow
InvMixColumn
RoundKeyAddtion
順序変換が可能
RoundKeyにInvMixColumn変換を行うことにより、順序変換が可能
復号化
RoundKey

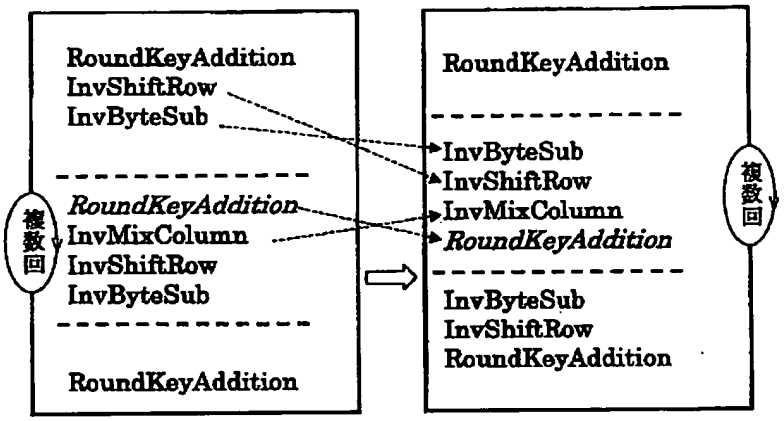
【図２】


データ変換処理
S100
暗号化と復号化との何れを行う？
復号化
暗号化
暗号化用RoundKeyを生成
S110
復号化用RoundKeyを生成
S140
各変換に用いる引数を暗号化用の引数に設定
S120
各変換に用いる引数を復号化用の引数に設定
S150
暗号化用のルックアップテーブル T0～T3を生成
S130
復号化用のルックアップテーブル T0～T3を生成
S160
RoundKeyAddition
S170
RoundFunction
S180
NO
RoundFunctionをNr-1回行った？
S190
YES
ByteSub
S200
ShiftRow
S210
RoundKeyAddition
S220
エンド

【図８】

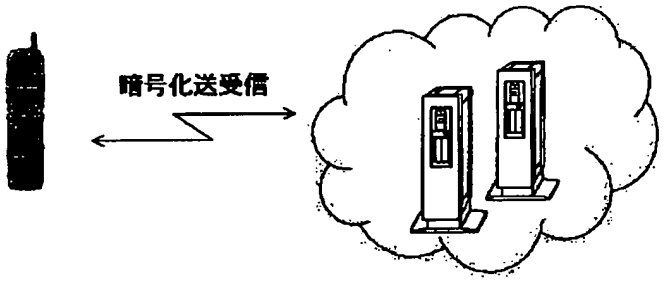

鍵サイズ:
128,192,256bit

【図５】



【図６】

実装結果



AES暗号クラスファイルサイズ：
2.4KB(Jarファイル) ＜ 10KB

(テスト用)就職先データベース